(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 742 522 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
13.11.1996 Bulletin 1996/46

(51) Int Cl.[6]: **G06F 13/26**

(21) Application number: **96302950.9**

(22) Date of filing: **26.04.1996**

(84) Designated Contracting States:
DE FR GB IT

(30) Priority: **12.05.1995 GB 9509626**

(71) Applicant: **SGS-THOMSON MICROELECTRONICS LTD.**
Bristol BS12 4SQ (GB)

(72) Inventor: **Simpson, Robert John**
Redland, Bristol BS6 6QD (GB)

(74) Representative: **Palmer, Roger et al**
PAGE, WHITE & FARRER
54 Doughty Street
London WC1N 2LS (GB)

(54) Processor interrupt control

(57) There is disclosed control circuitry for, and a method of controlling, multiple priority level interrupt request to a microprocessor in which output circuitry for outputting an interrupt identifier is operable only in response to an interrupt signal having a higher priority status than any currently executing interrupt process, and a microprocessor system and method of controlling a microprocessor system, incorporating such circuitry.

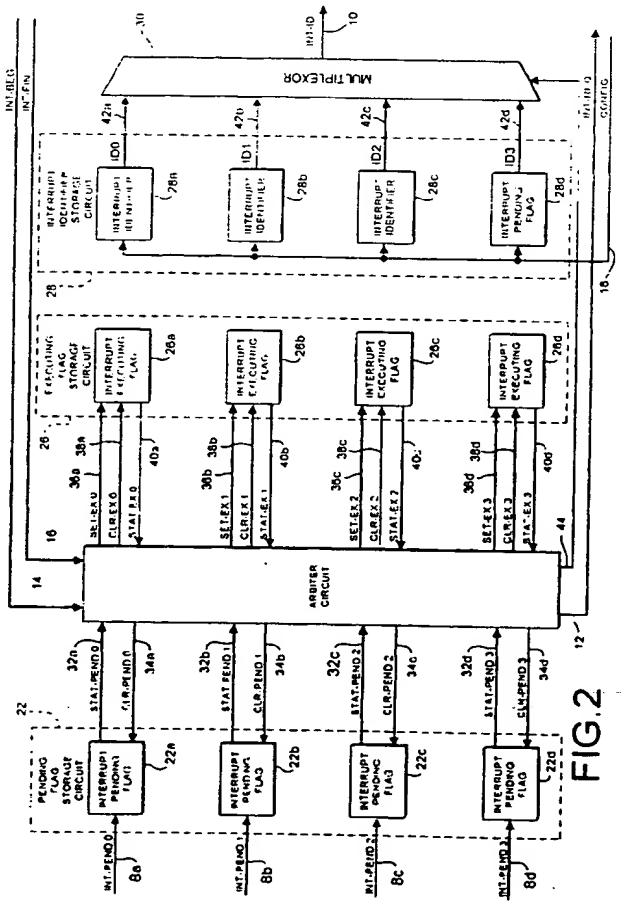

FIG.2

EP 0 742 522 A1

Description

The present invention relates to control of multiple priority level interrupt requests to a CPU of a microprocessor.

An interrupt is any action which causes the temporary cessation of execution by a processor of a particular process so that the processor can execute an interrupt process. Such action is typically initiated in processor circuits by peripheral devices requesting the processor to interrupt execution of a main process to execute an interrupt process associated with that particular peripheral device.

Known processors have been designed with a capacity to support a fixed number of interrupt priority levels. It may be necessary for a processor to support more than one interrupt since different interrupt requests require different interrupt processes. Not only do processors need to be able to support multiple interrupt requests, but they also need to be able to prioritise such requests should they occur simultaneously.

However, the number of interrupt priority levels which can be supported may be restricted by the processor. Typically a processor may support only two interrupt priority levels. Although processors may be adapted to support additional levels of interrupt, such adaptation is complex and it may be necessary to modify the instruction set of the processor and also to provide additional registers.

It is therefore an object of the present invention to provide an apparatus and a method for handling multiple priority level interrupt requests to a processor while reducing the problem of alteration to the processor itself.

According to one aspect of the present invention there is provided control circuitry for multiple priority level interrupt requests to a microprocessor comprising input circuitry having a plurality of parallel inputs for receiving respective interrupt signals and including storage circuitry to provide an indication of which inputs have received an interrupt signal and to retain said indication until an interrupt process associated with the respective interrupt signal has been completed, identifer storage circuitry for storing a plurality of interrupt identifiers, each interrupt identifier identifying an interrupt process for execution by the processor and corresponding to a respective one of said interrupt signals, arbiter circuitry connected to said storage circuitry for determining a priority status for each interrupt signal for which an indication is held by said storage circuitry and selecting the one interrupt signal with the highest priority status and output circuitry, operable to output an interrupt request signal only in response to an interrupt signal having a higher priority status than any currently executing interrupt process, and responsive to said arbiter circuitry for outputting a selected one of said plurality of interrupt identifiers associated with said one interrupt signal.

Thus there is provided circuitry for controlling interrupt requests to a processor which may be used for a large number of interrupt levels to be supported without modification to the processor. Preferably said output circuitry further outputs an interrupt request signal to the microprocessor.

It is also advantageous that interrupt processes can be interrupted by higher priority interrupt processes. Thus preferably said storage circuitry includes a first store for each input to indicate that an interrupt signal has been received for that input, and a second store for each input to indicate that the interrupt process of that input is currently executing.

According to a second aspect of the present invention there is provided a microprocessor system comprising a processor and a memory and including control circuitry for multiple priority level interrupt requests to the microprocessor, said control circuitry comprising input circuitry having a plurality of parallel inputs for receiving respective interrupt signals and including input storage circuitry to provide an indication of which inputs have received an interrupt signal and to retain said indication until an interrupt process associated with the respective interrupt signal has been completed, identifer storage circuitry for storing a plurality of interrupt identifiers, each interrupt identifier identifying an interrupt process for execution by the processor and corresponding to a respective one of said interrupt signals, arbiter circuitry connected to said storage circuitry for determining a priority status for each interrupt signal for which an indication is held by said storage circuitry and selecting the one interrupt signal with the highest priority status and output circuitry, operable only in response to an interrupt signal having a higher priority status than any currently executing interrupt process, and responsive to said arbiter circuitry for outputting a selected one of said plurality of interrupt identifiers asscciated with said one interrupt signal responsive to said arbiter circuitry for outputting a selected one of said plurality of interrupt identifiers associated with said one interrupt signal.

Preferably the memory has a plurality of allocated address spaces each associated with a respective one of said interrupt processes, wherein during execution of one of the interrupt processes parameters associated with an interrupted process are stored in one of the allocated address spaces. Thus additional priority levels may be added to the microprocessor system without alteration of the processor.

Preferably the processor, the memory and the control circuitry are formed on a single integrated circuit device.

According to a third aspect of the present invention there is for each interrupt signal for which an indication is stored, selecting the one interrupt signal with the highest priority status and outputting, in dependence on an interrupt signal having a higher priority status than any currently executing process, a selected one of said plurality of interrupt identifiers associated with said one interrupt provided a method of controlling multiple priority level interrupt requests to a microprocessor comprising

the steps of receiving at least one of a plurality of parallel interrupt signals, storing an indication of which interrupt signals have been received, retaining said indication until an interrupt process associated with the respective interrupt signal has been completed, storing a plurality of interrupt identifiers each interrupt identifier identifying an interrupt process for execution by the processor and corresponding to a respective one of said interrupt signals, determining a priority status for each interrupt signal for which an indication is stored, selecting the one interrupt signal with the highest priority status and outputting, in dependence on an interrupt signal having a higher priority status than any currently executing process, a selected one of said plurality of interrupt identifiers associated with said one interrupt signal.

Preferably the method further comprises the step of storing an indication of whether the interrupt process associated with an interrupt signal which has been received is currently executing.

Preferably the method further comprises the step of outputting an interrupt request signal to the microprocessor.

According to a fourth aspect of the present invention there is provided a method of controlling multiple priority level interrupt requests in a microprocessor system including a memory and a processor, said method comprising the steps of receiving at least one of a plurality of parallel interrupt signals, storing an indication of which interrupt signals have been received, retaining said indication until an interrupt process associated with the respective interrupt signal has been completed, storing a plurality of interrupt identifiers each interrupt identifier identifying an interrupt process for execution by the processor and corresponding to a respective one of said interrupt signals, determining a pricrity status signal, a selected one of said plurality of interrupt identifiers associated with said one interrupt signal.

Preferably the method further comprises the steps of allocating in the memory at least one address space for association with a respective interrupt process up to completion of the interrupt process, and storing in said at least one address space during execution of the interrupt process a parameter associated with a previous process which was interrupted by the interrupt process.

Preferably each executing interrupt process has allocated in memory a respective plurality of address spaces, each plurality of spaces holding parameters of a process which was interrupted by the respective interrupt process.

Preferably the interrupt identifiers are addresses of said memory.

An embodiment of the invention will now be described by way of example with reference to the accompanying drawings in which:

Figure 1 illustrates schematically an implementation of an interrupt controller according to a preferred embodiment of the present invention;

Figure 2 illustrates a detailed implementation of the interrupt controller of Figure 1; and

Figure 3 illustrates a reserved memory space associated with the interrupt controller according to the present invention.

Figure 1 illustrates an implementation of an interrupt controller 2 according to a preferred embodiment of the present invention in a circuit also including a microprocessor 4 and a memory 6.

The interrupt controller 2 receives in parallel four interrupt signals INT_PEND0 to INT_PEND3 on lines 8a to 8d. The interrupt controller also receives a signal INT_BEG on line 14 indicating that the processor has begun an interrupt process, a signal INT_FIN on line 16 indicating that the processor has finished an interrupt process, and a set of signals CONFIG on line 18 which are used to configure the interrupt controller. The interrupt controller outputs an interrupt request signal INT_REQ on line 12 and an interrupt identification signal INT_ID on line 10. The processor 4 receives the signal INT_REQ from the interrupt controller and outputs the signals INT_BEG, INT_FIN and CONFIG to the interrupt controller. The memory 6 receives the signal INT_ID from the interrupt controller. A set of control signals CONTROL on line 20 are provided between the processor 4 and the memory 6 to enable the processor 4 to control the memory 6.

The four interrupt signals INT_PEND0 to INT_PEND3 each originate from a dedicated interrupt source. When a particular interrupt source requires the processor to execute an interrupt process associated therewith, the interrupt source sets its associated interrupt signal so as to inform the interrupt controller that it requires service from the processor. The interrupt controller stores a plurality of interrupt identifers, in this example four interrupt identifiers, which identify an interrupt associated with a particular interrupt source. The number of interrupt identifers stored by the interrupt controller corresponds to the number of interrupt signals received by the interrupt controller. It will become apparent from the following description that the interrupt controller could be adapted to receive in parallel any number of interrupt signals from any number of interrupt sources, and thus store any number of interrupt identifiers for supporting any number of interrupt processes.

Prior to operation of the interrupt controller 2 it is configured by the processor 4 using the signal CONFIG. The processor loads into the interrupt controller each interrupt identifier associated with each interrupt process. The interrupt identifier will be a unique "tag" such as an address associated with a particular interrupt process which the processor can use to locate and execute the interupt process as will be described in detail hereinbelow. The signal CONFIG from the processor may also be used to determine the priority level to be associated with each interrupt process and thus to be associated with each of the interrupt signals

INT_PEND0 to INT_PEND3.

In response to one of the interrupt signals INT_PEND0 to INT_PEND3 being set the interrupt controller 2 sets the interrupt request signal INT_REQ to request the processor 4 to service an interrupt. At the same time the interrrupt controller outputs as the interrupt identification signal INT_ID the interrupt identifier associated with one of the interrupt signals INT_PEND0 to INT_PEND3 which has been set. In response to the interrupt request, the processor interrupts its current process and commences execution of the interrupt process associated with the interrupt identification signal INT_ID.

In a different embodiment of the invention, however, the interrupt identification signal INT_ID may also form the interrupt request signal to the processor, such an embodiment dispensing with the signal INT_REQ. In such an embodiment the processor may, for example, recognise an interrupt request by identifying a valid interrupt identification signal on line 10.

As will be described in detail hereinbelow, on interrupting a current process being performed by the processor to perform an interrupt process, it is necessary to store information associated with the interrupted process so that after executing the interrupt process the processor can return to, and continue executing, the interrupted process.

On commencing execution of the interrupt process the processor sets the signal INT_BEG so as to inform the interrupt controller that the interrupt is being serviced as requested. The interrupt controller is then aware that the particular interrupt which it had requested to be executed is being executed. Upon completion of execution of the interrupt process the processor sets the signal INT_FIN so that the interrupt controller is aware that the processor has finished execution of the interrupt and has returned to executing the interrupted process.

As each of the interrupt signals INT_PEND0 to INT_PEND3 is derived from a dedicated interrupt source, any one or all of the interrupt signals may be set simultaneously. The interrupt controller is therefore provided with arbiter circuitry which allocates a predetermined priority to each of the interrupt signals INT_PEND0 to INT_PEND3 and thus to each of the interrupt sources. The interrupt controller in the example of Figure 1 is a four priority level interrupt controller, but the interrupt controller could be expanded to handle any number of priority levels. Thus in the event of more than one of the interrupt signals INT_PEND0 to INT_PEND3 simultaneously being set, the interrupt controller will output as the interrupt identification signal INT_ID that interrupt identifier associated with one of the interrupt signals which have been set having the highest priority. After execution of that interrupt the interrupt controller will again request the processor to interrupt its process to perform the next highest priority interrupt associated with one of the interrupt signals which has been set having the next highest priority.

As will be described in further detail hereinbelow if, whilst the processor is executing an interrupt process, an interrupt signal associated with an interrupt having a higher priority is set, then the interrupt controller will request the processor to interrupt that interrupt process and perform the higher priority interrupt process. After executing the higher priority interrupt process the processor will return to executing the lower priority interrupt process, and therafter return to executing the interrupted process.

The operation of an interrupt controller according to a preferred embodiment of the invention will now be described in detail with reference to Figure 2, which shows an implementation of the interrupt controller 2 of Figure 1. In Figure 2 like numerals have been used to denote like parts of Figure 1 wherever possible.

The interrupt controller 2 of Figure 2 comprises a pending flag storage circuit 22 including four interrupt pending flags 22a to 22d, an arbiter circuit 24, an executing flag storage circuit 26 including four interrupt executing flags 26a to 26d, an interrupt identifier storage circuit 28 including four interrupt identifiers 28a to 28d, and a multiplexor 30. Each of the interrupt pending flags 22a to 22d receives a respective one of the interrupt signals INT_PEND0 to INT_PEND3 on lines 8a to 8d and a respective one of four clear interrupt pending signals CLR_PEND0 to CLR_PEND3 on lines 34a to 34d, and outputs four interrupt pending status signals STAT_PEND0 to STAT_PEND3 on lines 32a to 32d. The arbiter circuit 24 receives each of the interrupt pending status signals STAT_PEND0 to STAT_PEND3 and outputs each of the clear interrupt pending signals CLR_PEND0 to CLR PEND3. The arbiter circuit 24 also receives the signal INT_BEG on line 14 and the signal INT_FIN on line 16 and outputs the signal INT_REQ on line 12. The arbiter circuit 24 further ouputs four set interrupt executing signals SET_EX0 to SET_EX3 to respective ones of the interrupt executing flags 26a to 26d, four clear interrupt executing signals CLR_EX0 to CLR_EX3 to respective ones of the interrupt executing flags 26a to 26d, and a select interrupt identifier signal SEL_ID on line 44 to the multiplexor 30, and receives four interrupt executing status signals STAT_EX0 to STAT_EX3 on lines 40a to 40d from respective ones of the interrupt executing flags 26a to 26d. Each of the four interrupt identifiers 28a to 28d receive the signals CONFIG on line 18, and output a respective identifier signal ID0 to ID3 on lines 42a to 42d to the multiplexor 30. The multiplexor 30 is controlled by the signal SEL_ID to output one of the identifier signals ID0 to ID3 as the interrupt identifier signal INT_ID on line 10.

Each one of the four interrupt pending flags 22a to 22d, the four interrupt executing flags 26a to 26d, and the four interrupt identifiers 28a to 28d is associated with one of the four interrupt process priority levels.

Prior to operation of the interrupt controller, using the signals CONFIG, the processor loads into each of the interrupt identifiers 28a to 28d of the interrupt iden-

tifier storage circuit an identifier associated with each of the four interrupts. This interrupt identifier must be a unique "tag" which the processor can use to identify each particular interrupt. The interrupt identifier may be used to identify an instruction pointer, a pointer to an area of memory for use by the interrupt routine, or an area of memory in which parameters associated with the interrupted process can be stored. In the preferred embodiment the interrupt identifier is an address of main memory which is reserved for use by the interrupt. Each interrupt level has an area of main memory allocated for storage of parameters associated with an interrupted process, for example register contents of the processor associated with the interrupted process, which are essential for returning to the interrupted process. The use of the memory in this way allows extra interrupt levels to be added independently of the processor. The allocation of areas of main memory to interrupt levels will be explained with reference to Figure 3.

In a preferred embodiment the processor 4 is a transputer, and the interrupt identifier is a workspace address in main memory which is the starting address of a workspace associated with that particular interrupt process. Referring to Figure 3, stored in a fixed position offset from the workspace address is an interrupt instruction pointer which gives the address in main memory of the first instruction which is associated with that particular interrupt process. Other locations in the interrupt workspace offset from the interrupt process workspace address are used to store information associated with the interrupted process so that the interrupted process can be correctly returned to and executed after the interrupt has been executed. Thus if the processor 2 is a transputer then it is necessary to store the contents of a C register, a B register and an A register at the time of the interrupted process, the instruction pointer of the interrupted process, the workspace pointer of the interrupted process and the status of the interrupted process as shown in Figure 3.

It should also be noted that as explained hereinabove with reference to Figure 1, an interrupt process can itself be interrupted by an interrupt process. Thus it is possible that the main process has been interrupted by a level 0 interrupt, the level 0 interrupt has been interrupted by a level 1 interrupt, the level 1 interrupt has been interrupted by a level 2 interrupt, and the level 2 interrupt has been interrupted by a level 3 interrupt. In such a scenario the level 3 interrupt will be executing and upon completion the level 2, 1 and 0 interrupts will be successively executed, providing no other higher priority interrupts are pending, before returning to execution of the main process. Thus it is necessary to provide a reserved workspace area such as is shown in Figure 3 for each level of interrupt so that the interupted process can always be returned to after completion of any level of interrupt. Thus each interrupt identifier has a unique workspace address and reserved area for storing information associated with the interrupted process. It will be understood, however, that the interrupt identifer could be a different type of identification "tag". In particular the interrupt identifier could merely be the address of the first instruction of the interrupt process. If the processor is a standard CPU rather than a transputer, it will still be necessary to store information associated with the interrupted process before executing an interrupt process.

It is important to note that an interrupt process cannot be interrupted by an interrupt request from an interrupt source having the same priority level, and that an interrupt process must be completed and the interrupted process returned to before commencing execution of another interrupt process of the same priority level. As each interrupt priority level is preferably provided with its own workspace area which provides the identify of the interrupted process, then if another interrupt of the same level was commenced before returning to the lower level interrupt or the main process, the details of the interrupted process in the interrupt workspace would be over-written and it would be impossible to return to the interrupted process. Thus the processor must return to the interrupted process in the same cycle as the last instruction of the interrupt process to prevent such an eventuality.

The operation of the interrupt controller 2 of Figure 2 will now be described in detail. Each of the interrupt sources requesting to be serviced by the processor sets its corresponding interrupt signal INT_PEND0 to INT_PEND3, and this causes the respective ones of the interrupt pending flags 22a to 22b to be set. The arbiter circuit 24 cyclically monitors the status of the four interrupt pending flags 22a to 22b to check if any of them have been set by monitoring the interrupt pending status signals STAT_PEND0 to STAT_PEND3. On detecting that one or more of interrupt pending flags 22a to 22d has been set the arbiter circuit sets the interrupt request signal INT_REQ to the processor 2 and determines which of the interrupt pending flags set has the highest priority. The arbiter circuit then sets the signal SEL_ID to connect the appropriate one of the identifier signals ID1 to ID3 to the interrupt identifier signal INT_ID.

As described hereinbefore with reference to Figure 1 the processor then interrupts its current process and commences execution of the interrupt process identified by the interrupt identifier signal INT_ID. The processor sets the signal INT_BEG to indicate execution of the interrupt has begun. In response to the signal INT_BEG the arbiter circuit sets the appropriate one of the signals CLR_PEND0 to CLR_PEND3 to thereby clear the interrupt pending flag associated with the interrupt process now being executed, and also sets the appropriate one of the signals SET_EX0 to SET_EX3 to set the one of the interrupt executing flags 26a to 26d associated with that interrupt now being executed.

The arbiter circuit 24 thereafter continues to monitor the status of the interrupt pending flags 22a to 22d on the interrupt pending status signals STAT_PEND0 to

STAT_PEND3, and also monitors the status of the interrupt executing status signals STAT_EX0 to STAT_EX3 thereby to check which of the interrupt executing flags 26a to 26d are set. If an interrupt pending flag is set which is associated with an interrupt process having a higher priority than any currently executing interrupt process, then the currently executing interrupt process is interrupted by the interrupt controller 2 in the same manner as described hereinabove so that the higher priority interrupt process is executed. After executing the higher priority interrupt process the processor will return, under the control of the interrupt controller, to execute either an interrupted interrupt process or a pending interrupt depending on whichever has the higher priority.

The interrupt controller 2 may be provided on a single integrated circuit device with the processor 4 and the memory 6.

Although the interrupt controller according to the present invention has been described hereinabove with reference to a particular embodiment, it will be apparent to one skilled in the art how such an interrupt controller may be adapted for use with a wide variety of processors.

Claims

1. Control circuitry for multiple priority level interrupt requests to a microprocessor comprising:

   input circuitry having a plurality of parallel inputs for receiving respective interrupt signals and including input storage circuitry to provide an indication of which inputs have received an interrupt signal and to retain said indication until an interrupt process associated with the respective interrupt signal has been completed;
   identifer storage circuitry for storing a plurality of interrupt identifiers, each interrupt identifier identifying an interrupt process for execution by the processor and corresponding to a respective one of said interrupt signals;
   arbiter circuitry connected to said storage circuitry for determining a priority status for each interrupt signal for which an indication is held by said storage circuitry and selecting the one interrupt signal with the highest priority status; and
   output circuitry, operable only in response to an interrupt signal having a higher priority status than any currently executing interrupt process, and responsive to said arbiter circuitry for outputting a selected one of said plurality of interrupt identifiers associated with said one interrupt signal.

2. Control circuitry according to claim 1 wherein said output circuitry further outputs an interrupt request signal to the microprocessor.

3. Control circuitry according to claim 1 or claim 2 in which said storage circuitry includes a first store for each input to indicate that the interrupt signal has been received for that input, and a second store for each input to indicate that the interrupt process of that input is currently executing.

4. Control circuitry according to claim 3 comprising further input circuitry for receiving an interrupt commencement signal from the processor indicating the processor has begun execution of the interrupt process, said input circuitry being operably connected to said second store whereby an interrupt executing flag associated with said interrupt process is set by said commencement signal.

5. Control circuitry according to claim 4 in which said further input circuitry includes an input for receiving a completion signal from the processor indicating the processor has completed execution of the interrupt process, the interrupt executing flag associated with said interrupt process being reset by said completion signal.

6. Control circuitry according to claim 4 or claim 5 wherein said storage circuitry is connected to said further input circuitry so that said first store is reset by said commencement signal.

7. Control circuitry according to any preceding claim further comprising multiplexor circuitry connected to an output of the arbiter, wherein the multiplexor circuitry is controlled to output the appropriate one of said plurality of interrupt identifiers.

8. A microprocessor system comprising a processor and a memory and including control circuitry for multiple priority level interrupt requests to the microprocessor, said control circuitry comprising:

   input circuitry having a plurality of parallel inputs for receiving respective interrupt signals and including input storage circuitry to provide an indication of which inputs have received an interrupt signal and to retain said indication until an interrupt process associated with the respective interrupt signal has been completed;
   identifer storage circuitry for storing a plurality of interrupt identifiers, each interrupt identifier identifying an interrupt process for execution by the processor and corresponding to a respective one of said interrupt signals;
   arbiter circuitry connected to said storage circuitry for determining a priority status for each interrupt signal for which an indication is held

by said storage circuitry and selecting the one interrupt signal with the highest priority status; and

output circuitry, operable only in response to an interrupt signal having a higher priority status than any currently executing interrupt process, and responsive to said arbiter circuitry for outputting a selected one of said plurality of interrupt identifiers associated with said one interrupt signal.

9. A microprocessor system according to claim 8 wherein said output circuitry further outputs an interrupt request signal to the microprocessor.

10. A microprocessor system according to claim 8 or claim 9 in which said storage circuitry includes a first store for each input to indicate that the interrupt signal has been received for that input, and a second store for each input to indicate that the interrupt process of that input is currently executing.

11. A microprocessor system according to claim 10 comprising further input circuitry for receiving an interrupt commencement signal from the processor indicating the processor has begun execution of the interrupt process, said input circuitry being operably connected to said second store whereby an interrupt executing flag associated with said interrupt process is set by said commencement signal.

12. A microprocessor system according to claim 11 in which said further input circuitry includes an input for receiving a completion signal from the processor indicating the processor has completed execution of the interrupt process, the interrupt executing flag associated with said interrupt process being reset by said completion signal.

13. A microprocessor system according to claim 11 or claim 12 wherein said storage circuitry is connected to said further input circuitry so that said first store is reset by said commencement signal.

14. A microprocessor system according to any one of claims 8 to 13 further comprising multiplexor circuitry connected to an output of the arbiter, wherein the multiplexor circuitry is controlled to output the appropriate one of said plurality of interrupt identifiers.

15. A microprocessor system according to any one of claims 8 to 14 wherein the plurality of interrupt identifiers are addresses in said memory.

16. A microprocessor system according to any one of claims 8 to 15 wherein the memory has a plurality of allocated address spaces each associated with a respective one of said interrupt processes, wherein during execution of one of the interrupt processes parameters associated with an interrupted process are stored in one of the allocated address spaces.

17. A microprocessor system according to any one of claims 8 to 16 wherein the processor, the memory and the control circuitry are formed on a single integrated circuit device.

18. A method of controlling multiple priority level interrupt requests to a microprocessor comprising the steps of:

receiving at least one of a plurality of parallel interrupt signals;

storing an indication of which interrupt signals have been received;

retaining said indication until an interrupt process associated with the respective interrupt signal has been completed;

storing a plurality of interrupt identifiers each interrupt identifier identifying an interrupt process for execution by the processor and corresponding to a respective one of said interrupt signals;

determining a priority status for each interrupt signal for which an indication is stored;

selecting the one interrupt signal with the highest priority status; and

outputting, in dependence on an interrupt signal having a higher priority status than any currently executing process, a selected one of said plurality of interrupt identifiers associated with said one interrupt signal.

19. A method according to claim 18 further comprising the step of:

storing an indication of whether the interrupt process associated with an interrupt signal which has been received is currently executing.

20. A method according to claim 18 or claim 19 further comprising the step of outputting an interrupt request signal to the microprocessor.

21. A method according to claim 19 further comprising the step of:

receiving an interrupt commencement signal from the microprocessor indicating the microprocessor has begun commencement of the interrupt process, said indication of whether the interrupt process associated with an interrupt signal which has been received is currently executing being set in response to said interrupt commencement signal.

22. A method according to claim 21 further comprising the step of:

receiving a completion signal from the microprocessor indicating the microprocessor has com-

pleted execution of the interrupt process, said indication of whether the interrupt process associated with an interrupt signal which has been received is currently executing being reset in response to said completion signal.

23. A method according to claim 21 or claim 22 wherein the indication of whether an interrupt signal has been received is reset by said interrupt commencement signal.

24. A method according to any one of claims 18 to 23 wherein after executing an interrupt process the processor immediately returns to continue execution of the interrupted process.

25. A method of controlling multiple priority level interrupt requests in a microprocessor system including a memory and a processor, said method comprising the steps of:

receiving at least one of a plurality of parallel interrupt signals;
storing an indication of which interrupt signals have been received;
retaining said indication until an interrupt process associated with the respective interrupt signal has been completed;
storing a plurality of interrupt identifiers each interrupt identifier identifying an interrupt process for execution by the processor and corresponding to a respective one of said interrupt signals;
determining a priority status for each interrupt signal for which an indication is stored;
selecting the one interrupt signal with the highest priority status; and
outputting, in dependence on an interrupt signal having a higher priority status than any currently executing process, a selected one of said plurality of interrupt identifiers associated with said one interrupt signal.

26. A method according to claim 25 further comprising the step of:

storing an indication of whether the interrupt process associated with an interrupt signal which has been received is currently executing.

27. A method according to claim 25 or claim 26 further comprising the step of outputting an interrupt request signal to the microprocessor.

28. A method according to claim 26 further comprising the step of:

receiving an interrupt commencement signal from the microprocessor indicating the microprocessor has begun commencement of the interrupt process, said indication of whether the interrupt process associated with an interrupt signal which has been received is currently executing being set in response to said interrupt commencement signal.

29. A method according to claim 28 further comprising the step of:

receiving a completion signal from the microprocessor indicating the microprocessor has completed execution of the interrupt process, said indication of whether the interrupt process associated with an interrupt signal which has been received is currently executing being reset in response to said completion signal.

30. A method according to claim 28 or claim 29 wherein the indication of whether an interrupt signal has been received is reset by said interrupt commencement signal.

31. A method according to any one of claims 25 to 30 wherein after executing an interrupt process the processor immediately returns to continue execution of the interrupted process.

32. A method according to any one of claims 25 to 31 further comprising the steps of allocating in the memory at least one address space for association with a respective interrupt process up to completion of the interrupt process, and storing in said at least one address space during execution of the interrupt process a parameter associated with a previous process which was interrupted by the interrupt process.

33. A method according to claim 32 in which each executing interrupt process has allocated in memory a respective plurality of address spaces, each plurality of spaces holding parameters of a process which was interrupted by the respective interrupt process.

34. A method according to any one of claims 25 to 33 wherein the interrupt identifiers are addresses of said memory.

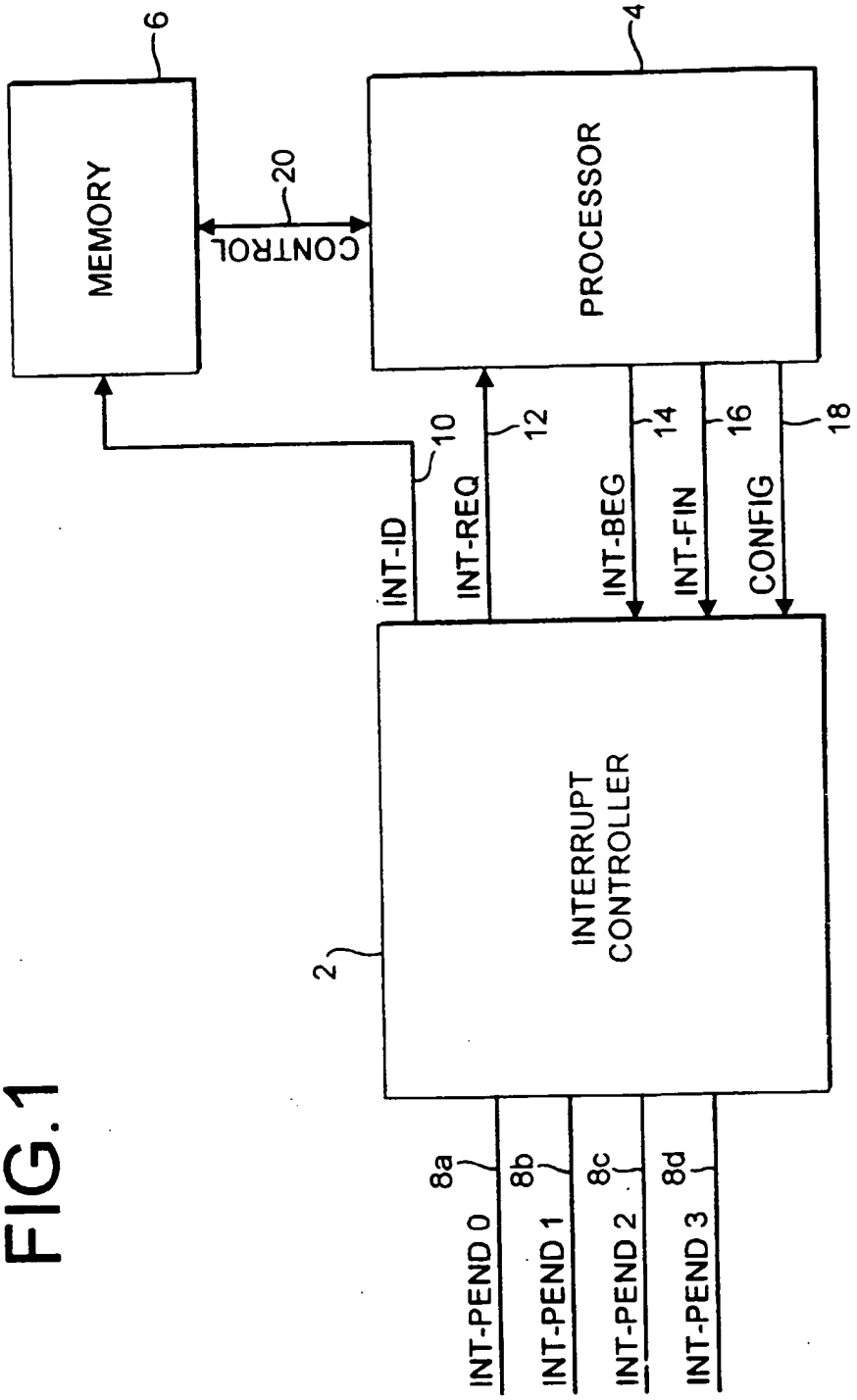
6
MEMORY
20
CONTROL
4
PROCESSOR
10
12
14
16
18
INT-ID
INT-REQ
INT-BEG
INT-FIN
CONFIG
2
INTERRUPT CONTROLLER
8a
8b
8c
8d
INT-PEND 0
INT-PEND 1
INT-PEND 2
INT-PEND 3

FIG.1

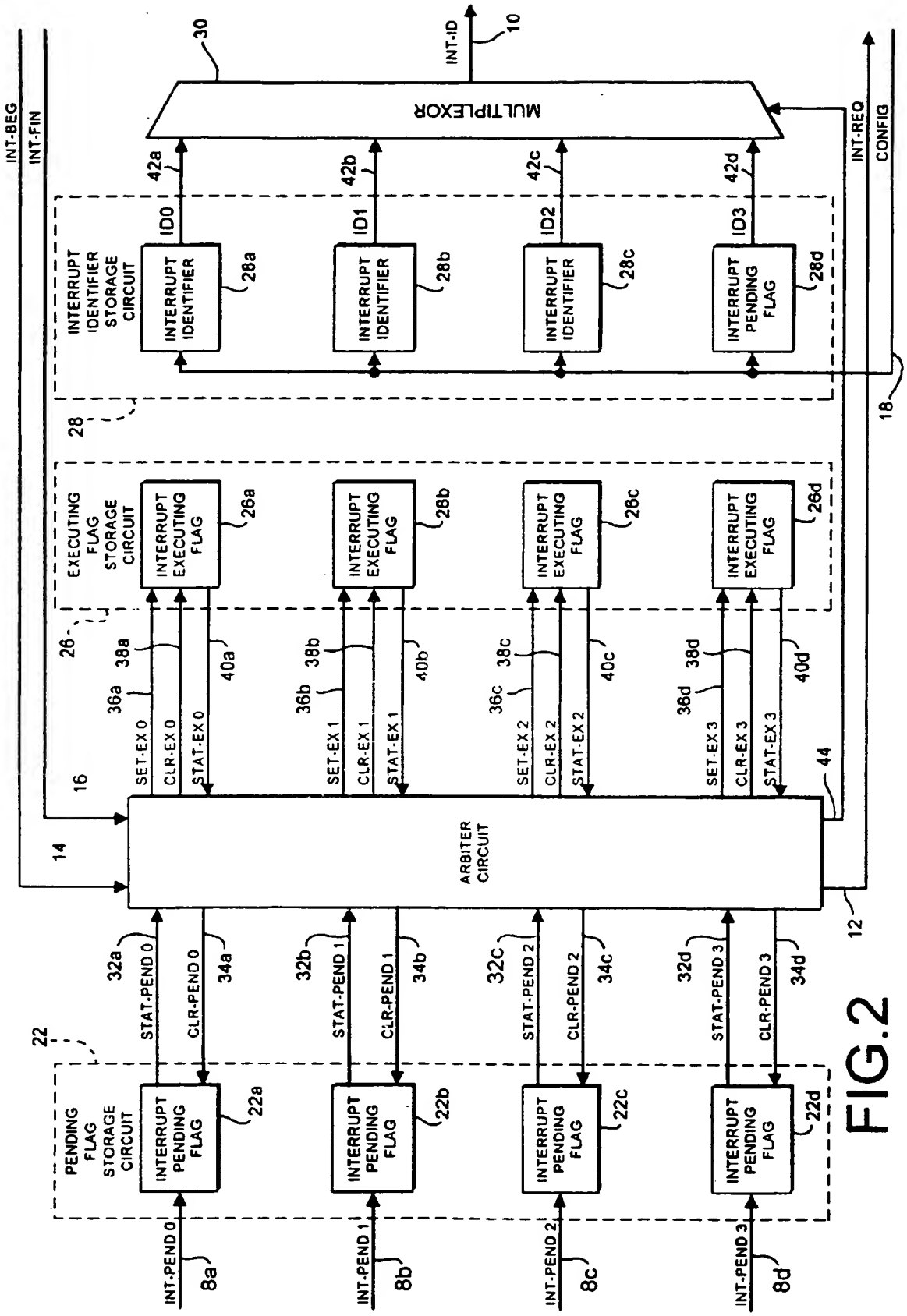
INT-BEG
INT-FIN
INT-ID
10
30
MULTIPLEXOR
INT-REQ
CONFIG
18
42a
42b
42c
42d
ID0
ID1
ID2
ID3
INTERRUPT IDENTIFIER STORAGE CIRCUIT
INTERRUPT IDENTIFIER
28a
28b
28c
INTERRUPT PENDING FLAG
28d
28
EXECUTING FLAG STORAGE CIRCUIT
INTERRUPT EXECUTING FLAG
26a
26b
26c
26d
26
38a
38b
38c
38d
40a
40b
40c
40d
36a
36b
36c
36d
SET-EX 0
CLR-EX 0
STAT-EX 0
SET-EX 1
CLR-EX 1
STAT-EX 1
SET-EX 2
CLR-EX 2
STAT-EX 2
SET-EX 3
CLR-EX 3
STAT-EX 3
16
44
14
ARBITER CIRCUIT
12
32a
32b
32c
32d
STAT-PEND 0
STAT-PEND 1
STAT-PEND 2
STAT-PEND 3
34a
34b
34c
34d
CLR-PEND 0
CLR-PEND 1
CLR-PEND 2
CLR-PEND 3
22
PENDING FLAG STORAGE CIRCUIT
INTERRUPT PENDING FLAG
22a
22b
22c
22d
INT-PEND 0
INT-PEND 1
INT-PEND 2
INT-PEND 3
8a
8b
8c
8d

FIG.2

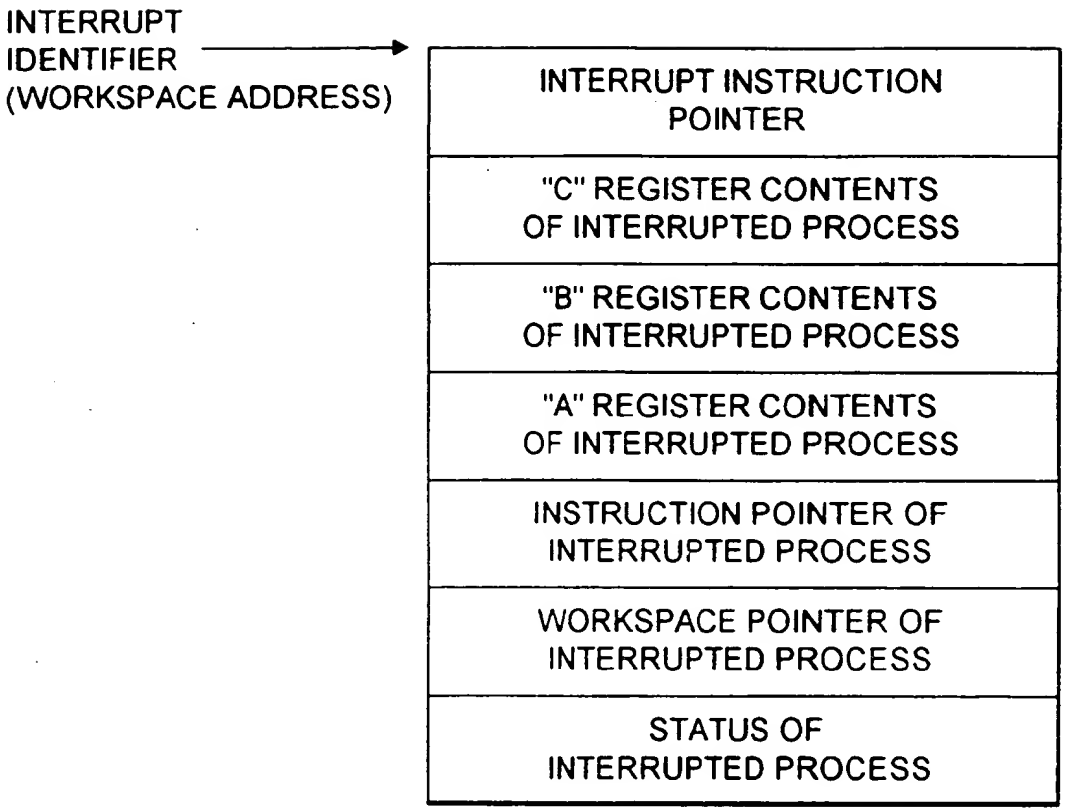
INTERRUPT
IDENTIFIER
(WORKSPACE ADDRESS)
INTERRUPT INSTRUCTION POINTER
"C" REGISTER CONTENTS OF INTERRUPTED PROCESS
"B" REGISTER CONTENTS OF INTERRUPTED PROCESS
"A" REGISTER CONTENTS OF INTERRUPTED PROCESS
INSTRUCTION POINTER OF INTERRUPTED PROCESS
WORKSPACE POINTER OF INTERRUPTED PROCESS
STATUS OF INTERRUPTED PROCESS

FIG.3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 96 30 2950

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	COMPUTER DESIGN, vol. 17, no. 9, September 1978, LITTLETON, MASSACHUSETTS US, pages 101-110, XP002012739 R. JASWA: "designing interrupt structures for multiprocessor systems" * page 108, left-hand column, line 6 - page 109, left-hand column, line 16; figure 7 *	1-34	G06F13/26
X	EP-A-0 652 514 (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.) * column 16, line 50 - column 18, line 22 *	1,8,18, 25	
A	EP-A-0 469 543 (GOLD STAR CO. LTD) * page 4, line 30 - line 33 *	1-34	
A	ELEKTRONIK, vol. 28, no. 10, May 1979, MUNCHEN DE, pages 73-80, XP002012740 H. WÄLTRING: "der Interrupt Controller - ein Baustein, der für Ordnung sorgt" * the whole document *	1-34	TECHNICAL FIELDS SEARCHED (Int.Cl.6) G06F
A	EP-A-0 581 480 (ADVANCED MICRO DEVICES, INC.) * page 3, line 17 - line 35; figure 4 *	1-34	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	5 September 1996	Jones, H

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

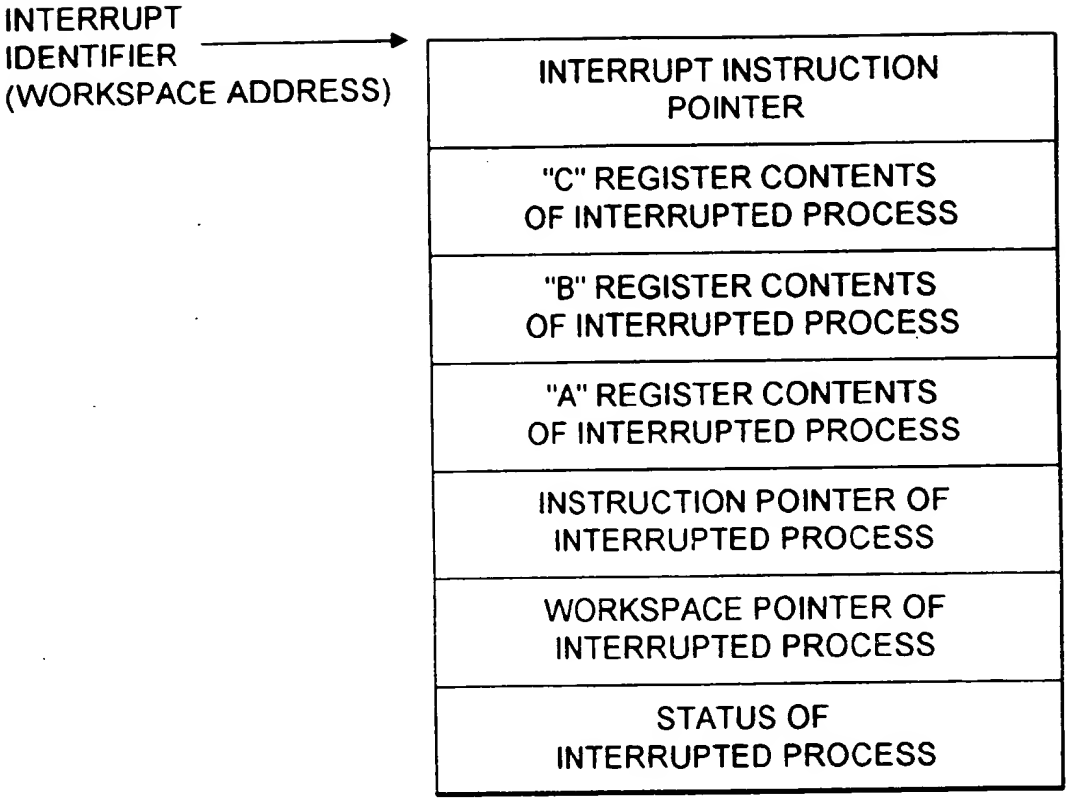
INTERRUPT
IDENTIFIER
(WORKSPACE ADDRESS)
INTERRUPT INSTRUCTION POINTER
"C" REGISTER CONTENTS OF INTERRUPTED PROCESS
"B" REGISTER CONTENTS OF INTERRUPTED PROCESS
"A" REGISTER CONTENTS OF INTERRUPTED PROCESS
INSTRUCTION POINTER OF INTERRUPTED PROCESS
WORKSPACE POINTER OF INTERRUPTED PROCESS
STATUS OF INTERRUPTED PROCESS

FIG.3

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	PATENT ABSTRACTS OF JAPAN vol. 015, no. 363 (P-1251), 12 September 1991 & JP 03 139751 A (TOSHIBA CORP;OTHERS: 01), 13 June 1991 see abstract	12
A	US 4 768 149 A (KONOPIK BRADLY J ET AL) 30 August 1988 see column 6, line 10 - column 7, line 31	1-12
A	US 5 218 703 A (FLECK ROD ET AL) 8 June 1993 cited in the application see the whole document	1-12

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0306042 A	08-03-1989	AU 2187088 A	09-03-1989
		CA 1307852 A	22-09-1992
		DE 3889267 D	01-06-1994
		DE 3889267 T	08-12-1994
		DK 492188 A	04-03-1989
		FI 884026 A	04-03-1989
		JP 1094466 A	13-04-1989
		MX 171367 B	21-10-1993
		US 5274825 A	28-12-1993
US 5381552 A	10-01-1995	NONE	
US 5287523 A	15-02-1994	NONE	
US 4768149 A	30-08-1988	CA 1251869 A	28-03-1989
		DE 3687866 A	08-04-1993
		EP 0212393 A	04-03-1987
		JP 2019400 C	19-02-1996
		JP 7054500 B	07-06-1995
		JP 62052655 A	07-03-1987
US 5218703 A	08-06-1993	DE 58908227 D	29-09-1994
		EP 0349905 A	10-01-1990
		JP 2059848 A	28-02-1990
		JP 2767751 B	18-06-1998